MANUAL DE USUARIO

SOFTWARE QUE RECOPILA LOS DIFERENTES MÉTODOS DE **CRIPTOGRAFÍA**

Seguridad Informática

INTEGRANTES:

- ALCANTARA ZUÑIGA, Alex Rodolfo
- PANTA PISCOCHE, Jose Diego
- RAMOS ENCARNACION, Nilton
- TREJO OBREGON, Rodrigo Emilio
- VASQUEZ RAMOS, Jose Manuel

UNS
UNIVERSIDAD
NACIONAL DEL SANTA

UNIVERSIDAD NACIONAL DEL SANTA

FACULTAD DE INGENIERÍA

Escuela Profesional de Ingeniería de Sistemas e Informática

SEGURIDAD INFORMÁTICA

IX CICLO – 2025 01

SOFTWARE QUE RECOPILA LOS DIFERENTES MÉTODOS DE CRIPTOGRAFÍA

INTEGRANTES:

- ALCANTARA ZUÑIGA, Alex Rodolfo
- PANTA PISCOCHE, Jose Diego
- RAMOS ENCARNACION, Nilton
- TREJO OBREGON, Rodrigo Emilio
- VASQUEZ RAMOS, Jose Manuel

DOCENTE:

Dr. GIL ALBARRAN GUILLERMO EDWARD

NUEVO CHIMBOTE – PERÚ

Julio, 2025

ÍNDICE DE CONTENIDO

I. INSTALACIÓN

1.1. Requisitos Previos

- **Python:** Puedes verificar si tienes Python instalado ejecutando `python --version` en la terminal. Si no, puedes descargarlo desde https://www.python.org/downloads/
- **Git:** Para clonar el repositorio. Puedes verificar su instalación con `git --version`. Descárgalo desde https://git-scm.com/downloads
- **pip:** El gestor de paquetes de Python. Generalmente viene instalado con Python.

Una vez verificados los requisitos, sigue estos pasos detallados para ejecutar el proyecto en tu máquina local.

1.2. Clonar el Repositorio

Lo primero que debemos hacer es clonar el repositorio desde GitHub para obtener los archivos del proyecto en tu máquina local. Abre tu terminal o línea de comandos y ejecuta el siguiente comando:

```
git clone https://github.com/josevasquezramos/seguridadinformatica_criptografia.git
```

Este comando descargará una copia completa del proyecto en tu directorio actual.

Después de clonar el repositorio, ingresa al directorio del proyecto con el siguiente comando:

```
cd seguridadinformatica_criptografia
```

Esto cambiará el directorio de trabajo a la carpeta del proyecto.

1.3. Crear un Entorno Virtual

El siguiente paso es crear un entorno virtual, lo cual es una buena práctica en desarrollo de software, ya que te permite tener una instalación de Python aislada con las dependencias necesarias para este proyecto sin interferir con otras configuraciones en tu sistema.

Ejecuta el siguiente comando para crear un entorno virtual dentro del proyecto:

```
python -m venv venv
```

Este comando creará una carpeta llamada venv dentro de tu proyecto, donde se almacenarán todas las dependencias de Python específicas para este proyecto.

1.4. Activar el Entorno Virtual

Una vez creado el entorno virtual, necesitamos activarlo para poder usar las librerías de Python que se instalarán en él.

- **En Windows:**

  Para activar el entorno virtual en Windows, utiliza el siguiente comando:

  ```
  venv\Scripts\activate
  ```

  Verás que el nombre del entorno virtual (venv) aparece al inicio de la línea de comandos, lo que indica que está activo.

- **En Linux/Mac:**

  En sistemas basados en Linux o Mac, el comando para activar el entorno es:

  ```
  source venv/bin/activate
  ```

  Al igual que en Windows, verás que el nombre del entorno (venv) aparece en tu terminal, confirmando que el entorno virtual está activo.

1.5. Instalar las Dependencias

Una vez el entorno virtual esté activo, necesitas instalar las dependencias necesarias para que el proyecto funcione correctamente. Las dependencias están listadas en el archivo requirements.txt del proyecto.

Ejecuta el siguiente comando para instalar todas las dependencias:

```
pip install -r requirements.txt
```

Este comando descargará e instalará todas las bibliotecas necesarias como se especifica en el archivo requirements.txt.

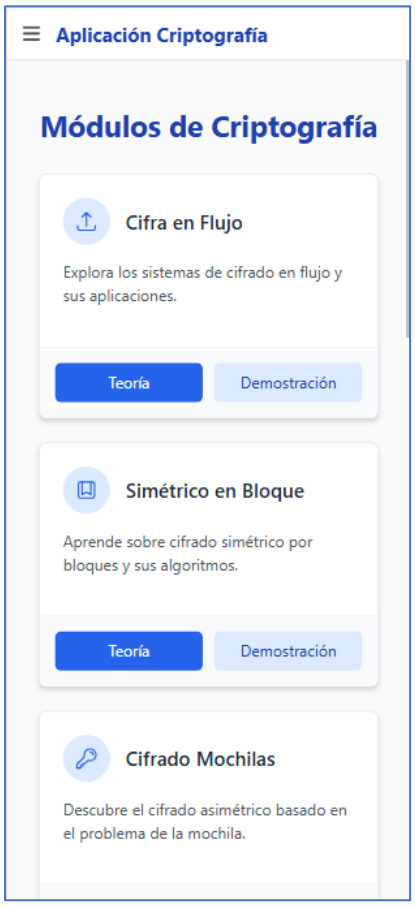
1.6. Ejecutar la Aplicación

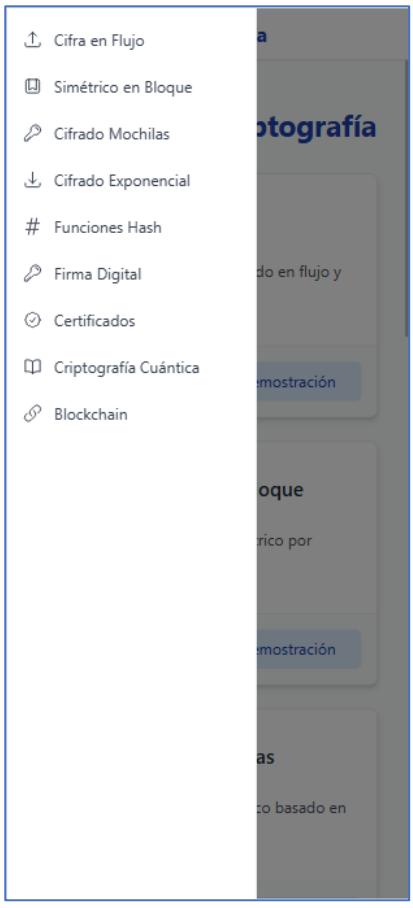
Con todas las dependencias instaladas, ya estás listo para ejecutar la aplicación. En la terminal, escribe el siguiente comando para iniciar el servidor de la aplicación:

```
python criptografia/app.py
```

Esto ejecutará el archivo principal de la aplicación (app.py) ubicado en la carpeta criptografia, y podrás ver la aplicación corriendo localmente en tu máquina.

Una vez que la aplicación esté corriendo localmente, abre tu navegador web y accede al siguiente enlace: http://127.0.0.1:8080/





II. RESUMEN TEÓRICO DE LOS MÓDULOS DEL SISTEMA

2.1. Sistemas de Cifra en Flujo

Un sistema de cifra en flujo es un tipo de algoritmo criptográfico que cifra los datos en bits o en caracteres de forma continua, a medida que se reciben. A diferencia del cifrado en bloque, que cifra datos en bloques fijos, en los sistemas de cifra en flujo, se utiliza una clave generada de forma pseudo-aleatoria para transformar los datos, en general en tiempo real. Son especialmente útiles para aplicaciones donde los datos llegan de forma continua, como en la transmisión de voz o video.

2.2. Cifrado Simétrico en Bloque

El cifrado simétrico en bloque es un tipo de cifrado en el cual los datos son cifrados en bloques de longitud fija, utilizando la misma clave tanto para cifrar como para descifrar. Un ejemplo popular de cifrado simétrico en bloque es el AES (Advanced Encryption Standard). Este método es eficiente y seguro, pero la principal desventaja es que ambas partes deben compartir y proteger la misma clave secreta.

2.3. Cifrado Asimétrico con Mochilas

El cifrado asimétrico con mochilas es un sistema de cifrado basado en un problema matemático relacionado con las mochilas, en el cual se resuelve un conjunto de sumas o combinaciones. Cada usuario tiene un par de claves, una pública y una privada, y la seguridad del sistema se basa en la dificultad de resolver el problema de la mochila. Este tipo de cifrado se utiliza principalmente en sistemas de clave pública, como el algoritmo RSA.

2.4. Cifrado Asimétrico Exponencial

El cifrado asimétrico exponencial se refiere a sistemas de cifrado en los que las claves se basan en operaciones de exponentes y logaritmos. Un ejemplo común de este tipo de cifrado es el algoritmo RSA. En este sistema, las claves pública y privada son matemáticamente diferentes, pero relacionadas. La seguridad se basa en la dificultad de resolver el problema de la factorización de grandes números.

2.5. Funciones Hash en Criptografía

Las funciones hash son algoritmos que transforman una entrada de datos de longitud variable en una cadena de longitud fija, generalmente un número o una secuencia de caracteres. Estas funciones son unidireccionales (es decir, no se puede obtener la entrada original a partir del

hash) y se utilizan ampliamente en criptografía para verificar la integridad de los datos, generar firmas digitales y en la construcción de contraseñas seguras.

2.6. Autenticación y Firma Digital

La autenticación y la firma digital son procesos que garantizan la identidad y la integridad de un mensaje o documento. La autenticación asegura que un mensaje proviene de la persona que dice ser, y la firma digital es una forma de verificación criptográfica de la autenticidad y no alteración del mensaje. En este proceso, se utilizan claves privadas para firmar el mensaje y claves públicas para verificar la firma.

2.7. Certificados Digitales

Un certificado digital es un documento electrónico que autentica la identidad de una persona, organización o dispositivo en una red. Contiene información como el nombre del titular, la clave pública y la firma de una autoridad de certificación (CA). Su principal función es garantizar que las claves públicas en el certificado corresponden realmente a la persona o entidad que se dice ser, permitiendo un intercambio de información seguro.

2.8. Criptografía Cuántica

La criptografía cuántica es un campo emergente de la criptografía que utiliza principios de la mecánica cuántica para mejorar la seguridad de las comunicaciones. Se basa en el principio de superposición y entrelazamiento cuántico para proteger la información de interferencias y observaciones externas, haciendo que cualquier intento de espiar una transmisión cambie el estado de los datos, alertando de una posible amenaza. Un ejemplo conocido es el protocolo BB84 para el intercambio seguro de claves.

2.9. Blockchain

Blockchain es una tecnología de registro descentralizado que permite la creación de un libro de contabilidad digital, distribuido entre varios nodos, de manera que ninguna entidad central tiene control total sobre los registros. Los bloques de datos, que contienen transacciones, están vinculados entre sí mediante técnicas criptográficas, lo que hace que sean casi imposibles de alterar una vez añadidos. Blockchain se utiliza principalmente en criptomonedas, como Bitcoin, pero también tiene aplicaciones en áreas como contratos inteligentes y gestión de cadenas de suministro.

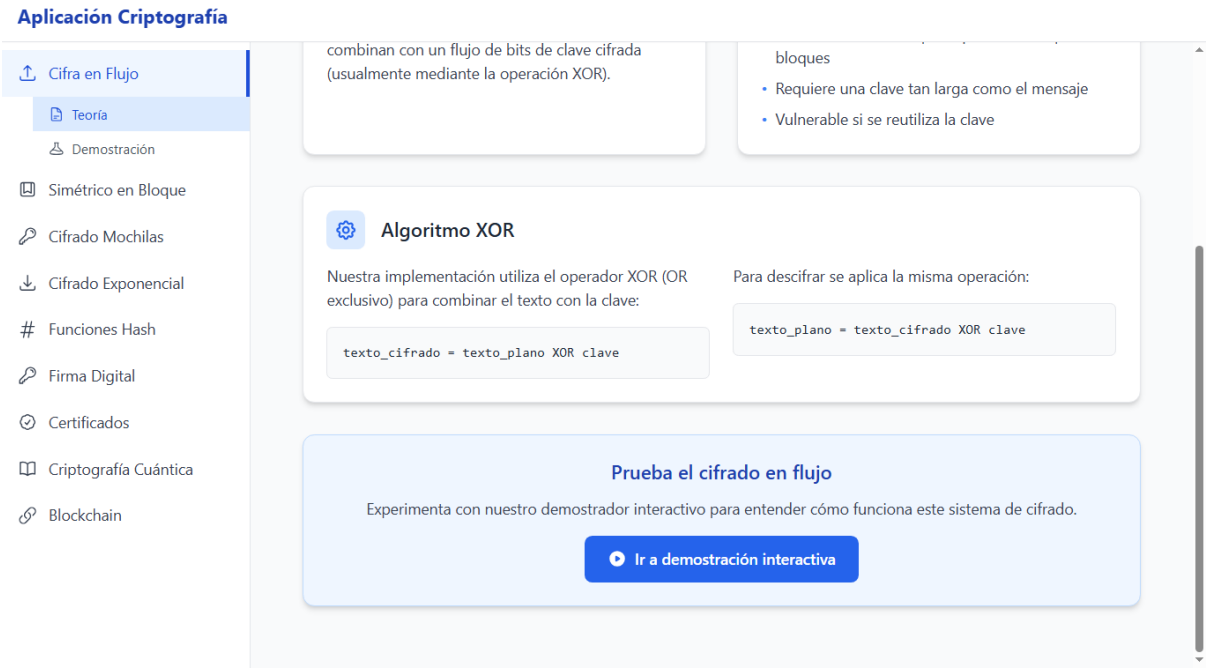
III. DEMOSTRACIÓN DE LA APLICACIÓN

3.1. Página de Inicio

Aquí podremos encontrar el índice de todos los módulos y links a las páginas de teorías y demostraciones de los algoritmos.

3.2. Cifrado en flujo

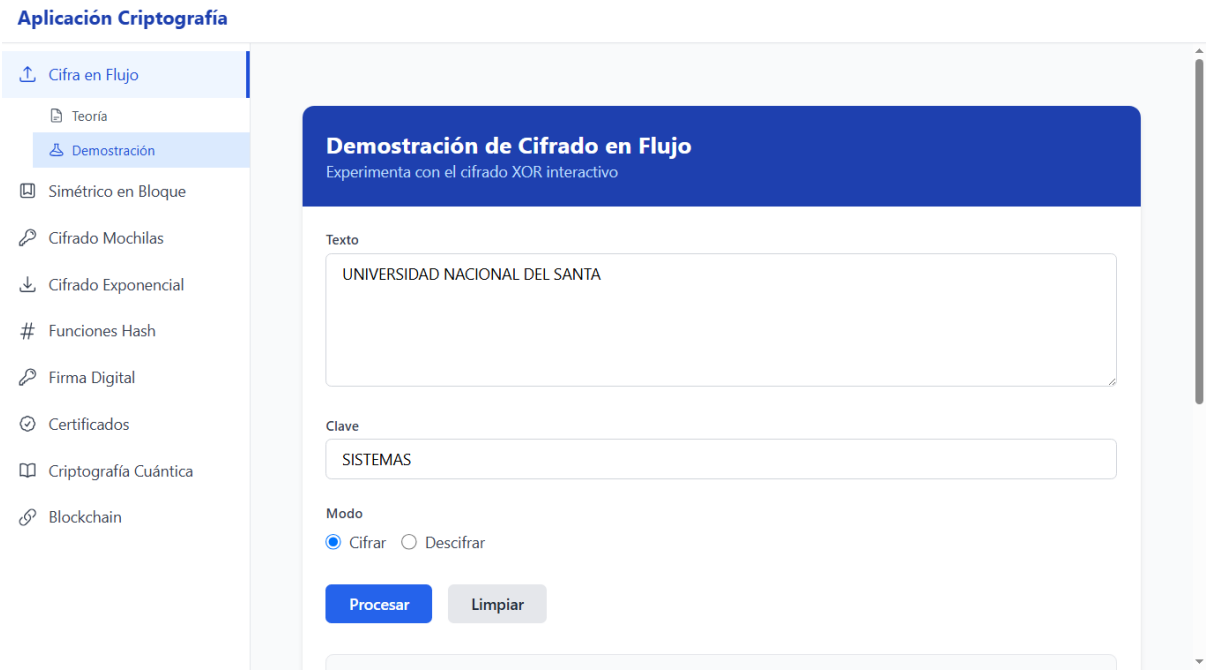
Teoría:



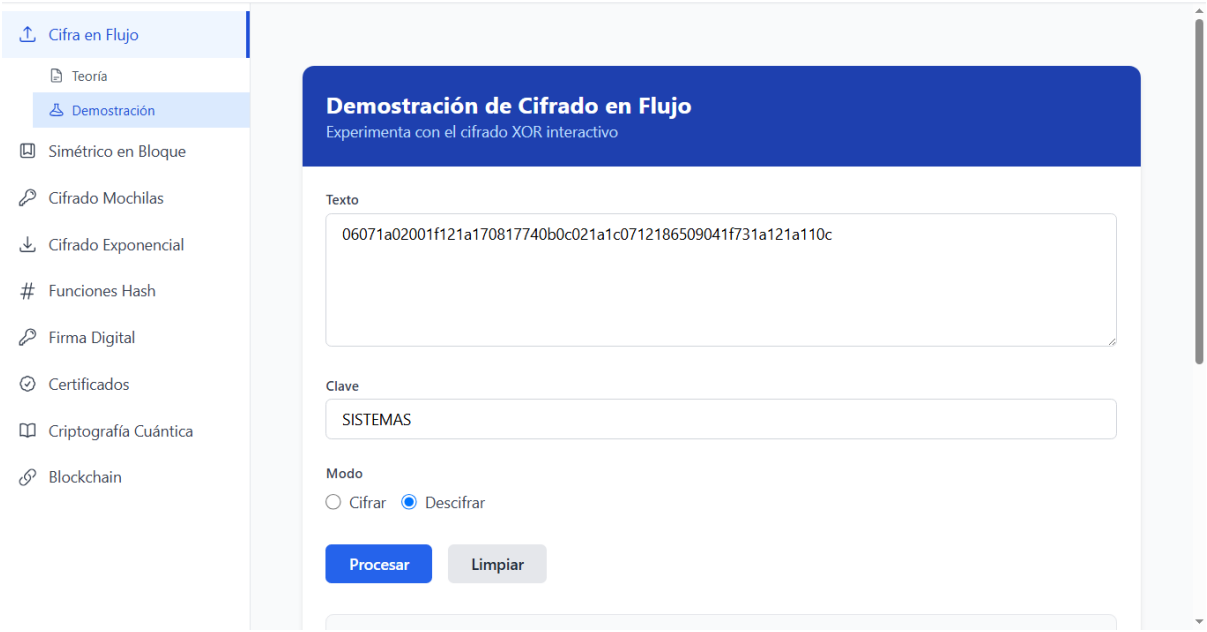
Demostración:

Instrucciones

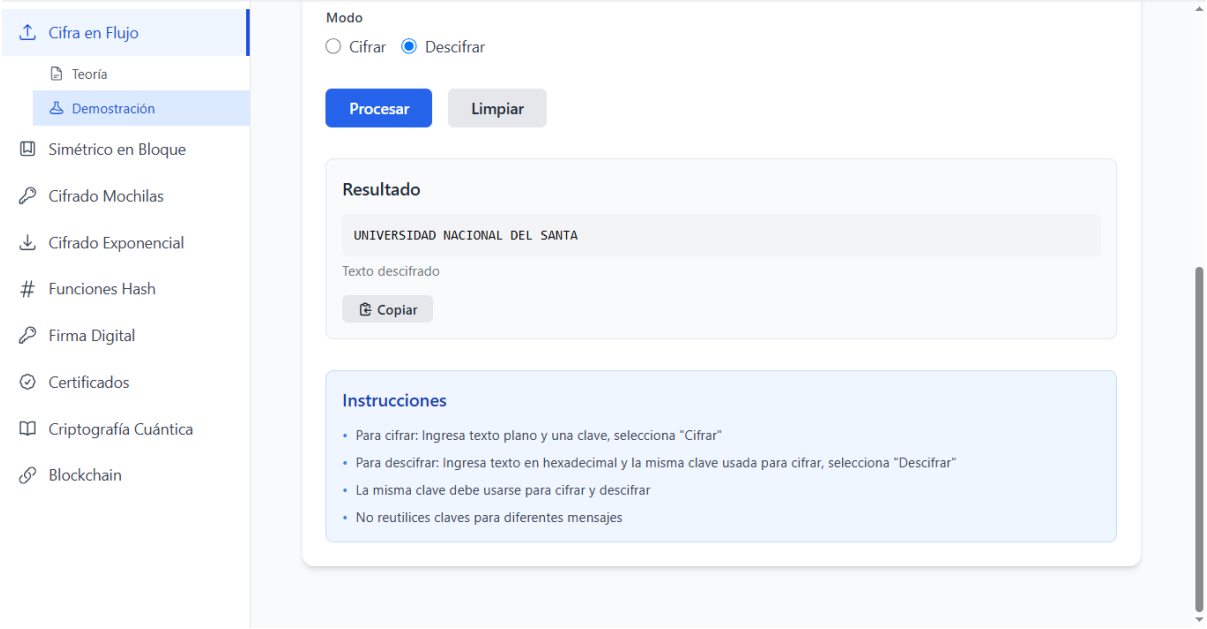
- Para cifrar: Ingresa texto plano y una clave, selecciona "Cifrar"
- Para descifrar: Ingresa texto en hexadecimal y la misma clave usada para cifrar, selecciona "Descifrar"
- La misma clave debe usarse para cifrar y descifrar
- No reutilices claves para diferentes mensajes



Aplicación Criptografía

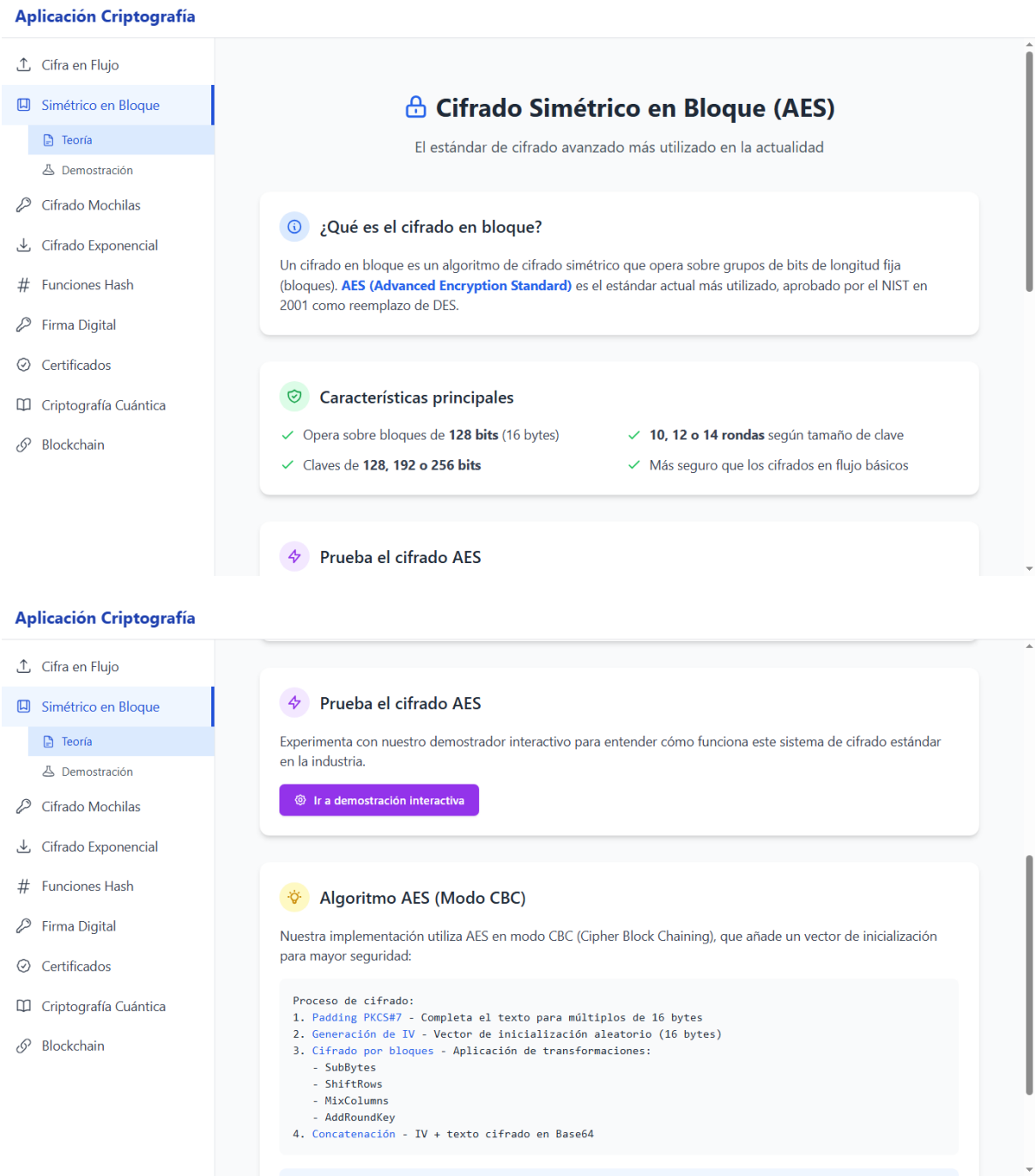


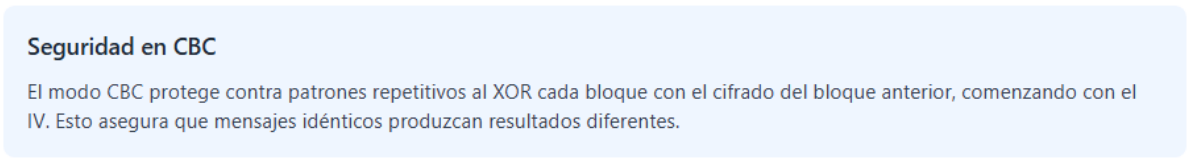
Aplicación Criptografía



3.3. Cifrado Simétrico en Bloque

Teoría:

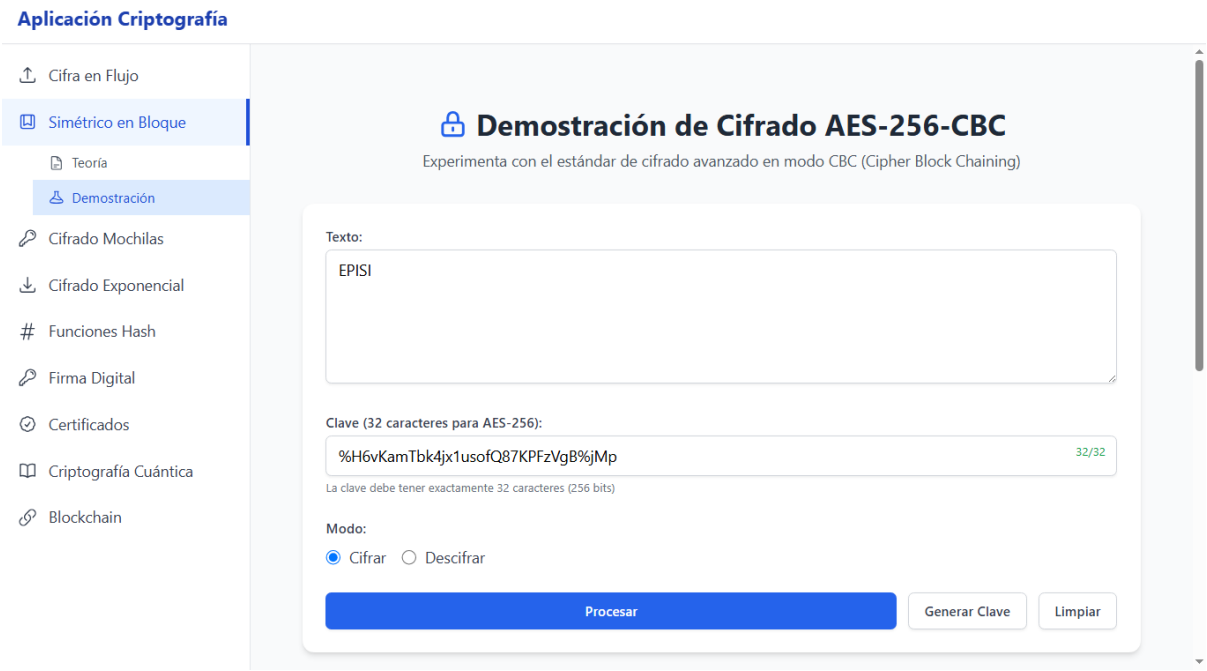


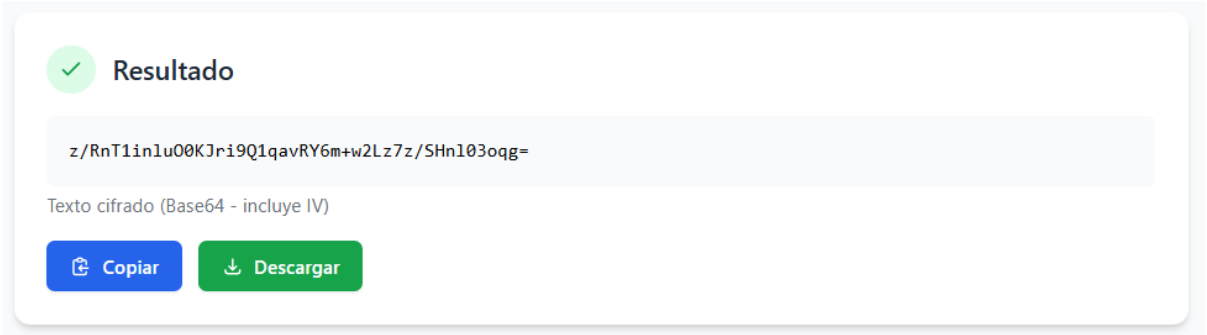


Demostración:

Instrucciones

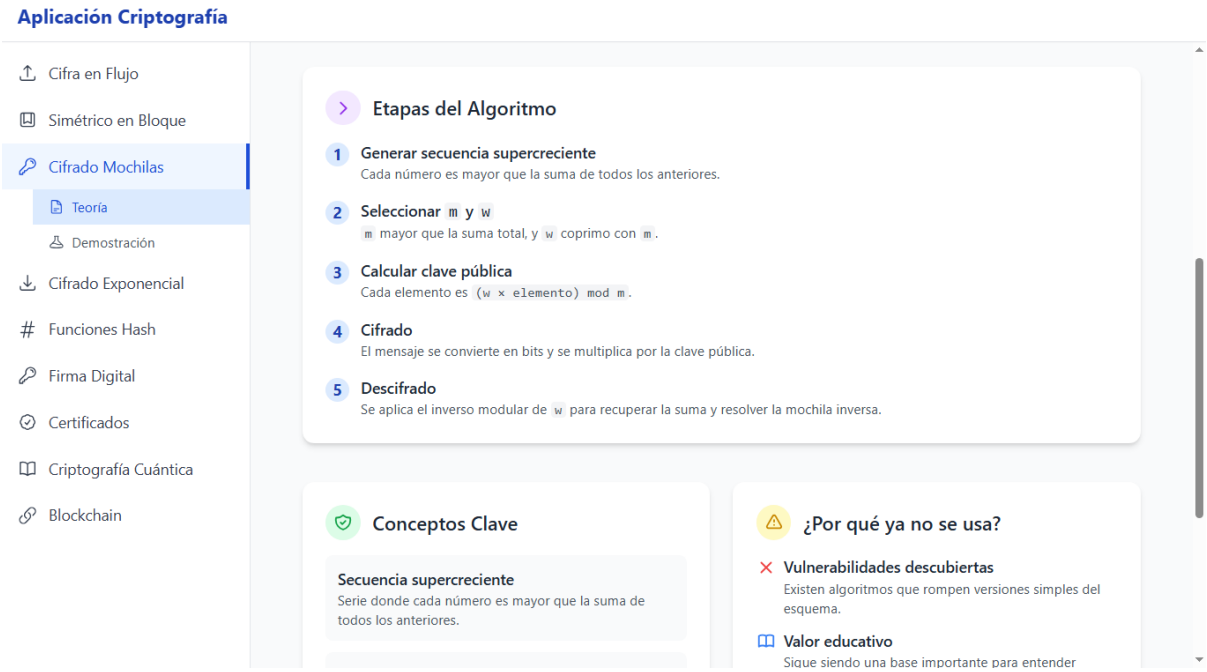
- Para cifrar: Ingresa texto plano y una clave de exactamente 32 caracteres
- Para descifrar: Ingresa texto en Base64 (incluye IV) y la misma clave usada para cifrar
- El sistema usa AES-256-CBC con padding PKCS7
- El IV (Vector de Inicialización) se genera automáticamente y se incluye en el resultado cifrado
- Guarda la clave de forma segura, es esencial para el descifrado

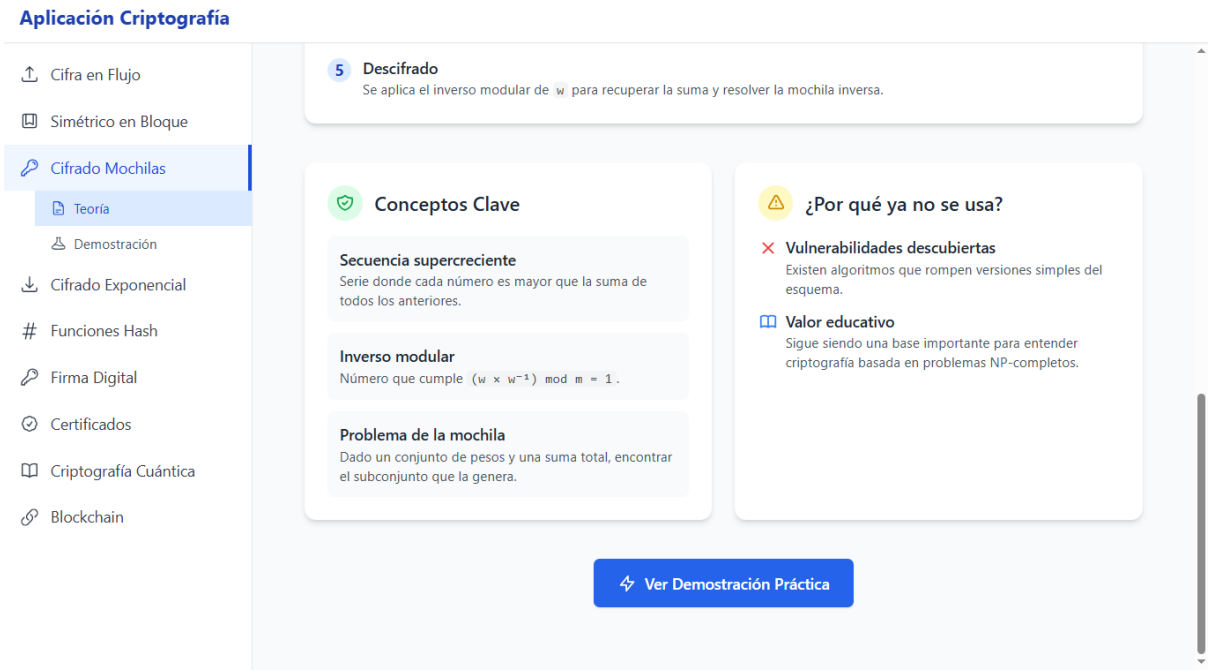




3.4. Cifrado Asimétrico con Mochilas

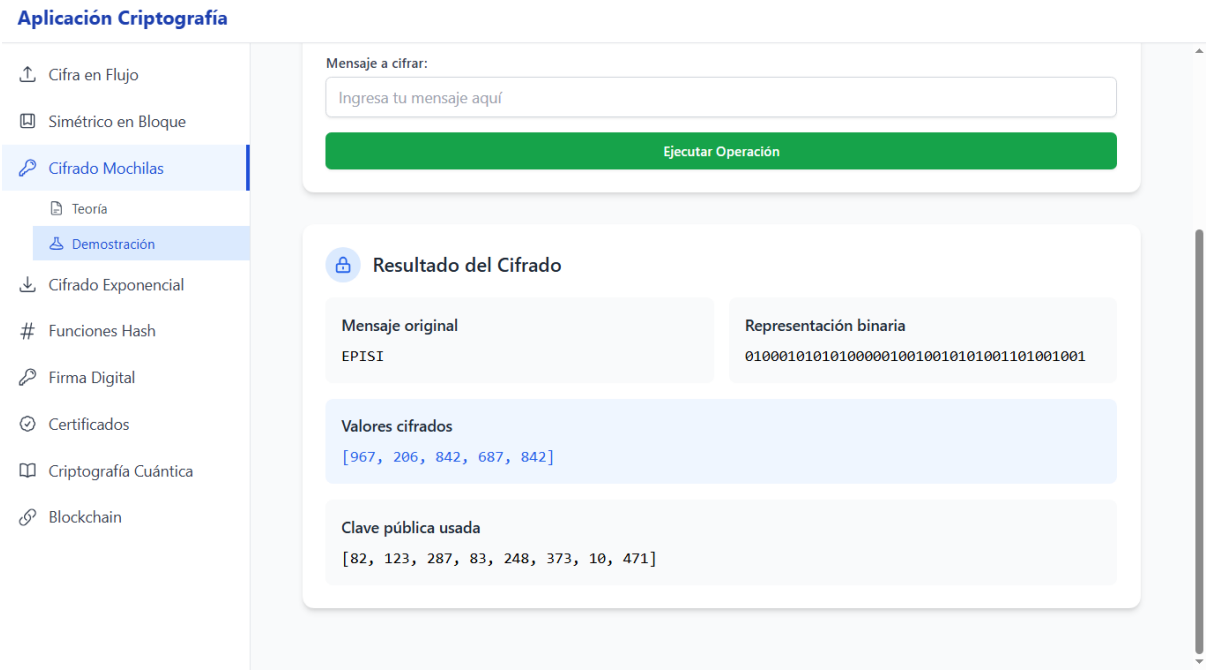
Teoría:





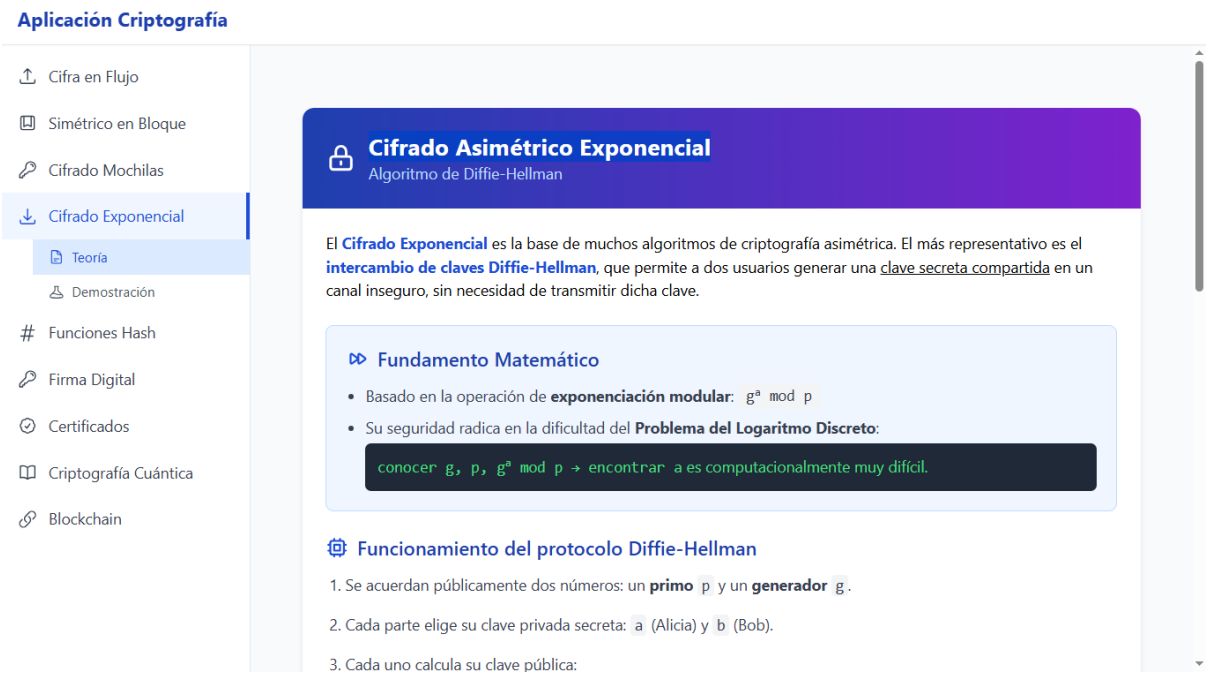
Demostración:

Ingrese un mensaje y de clic en Ejecutar operación.

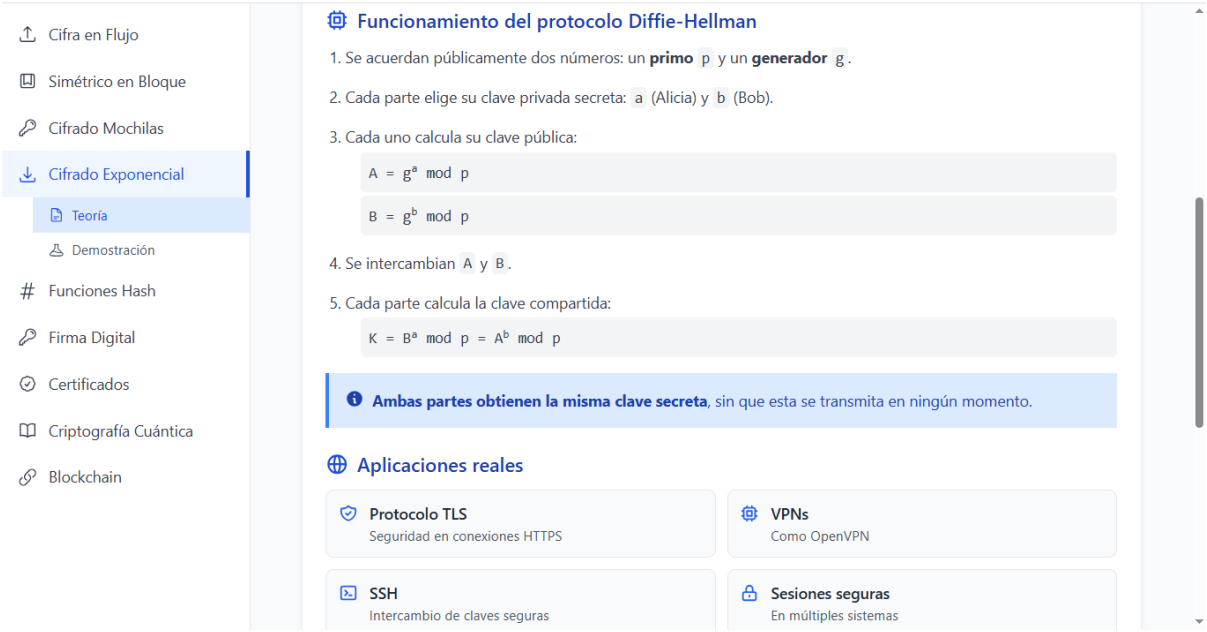


3.5. Cifrado Asimétrico Exponencial

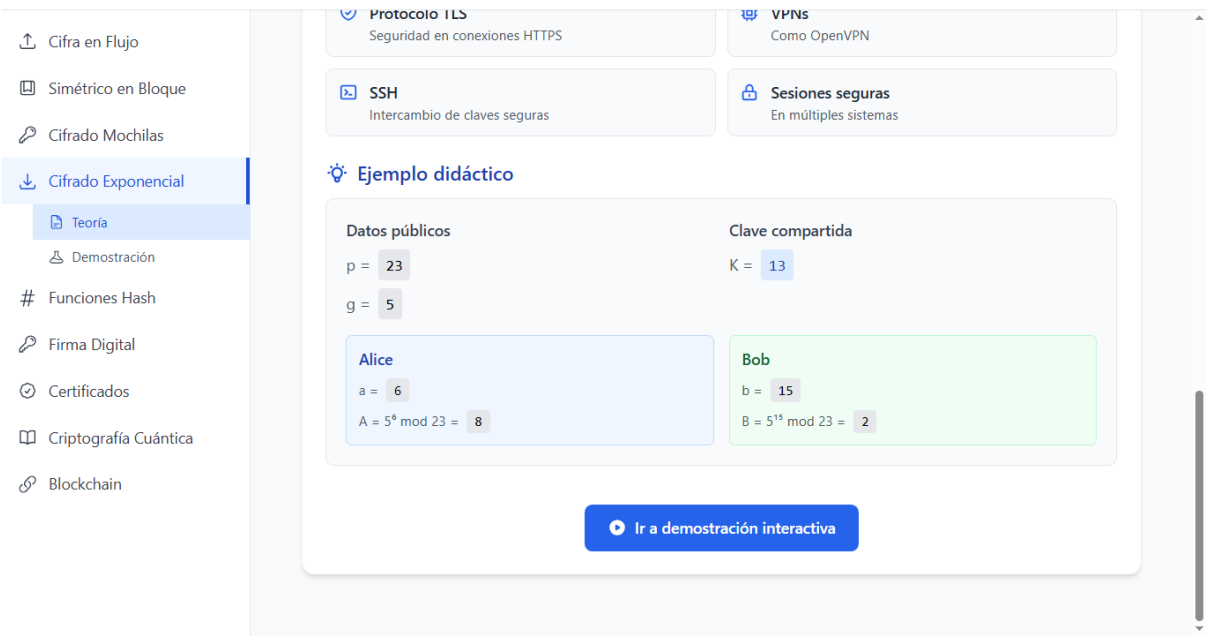
Teoría:



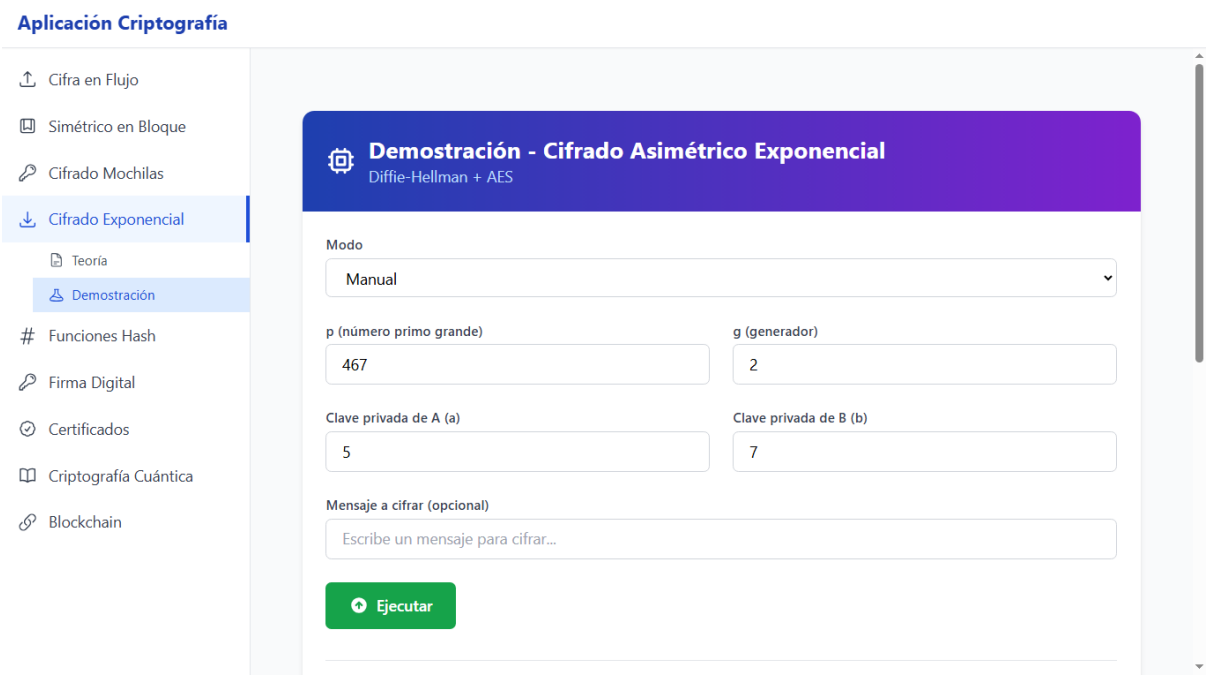
Aplicación Criptografía

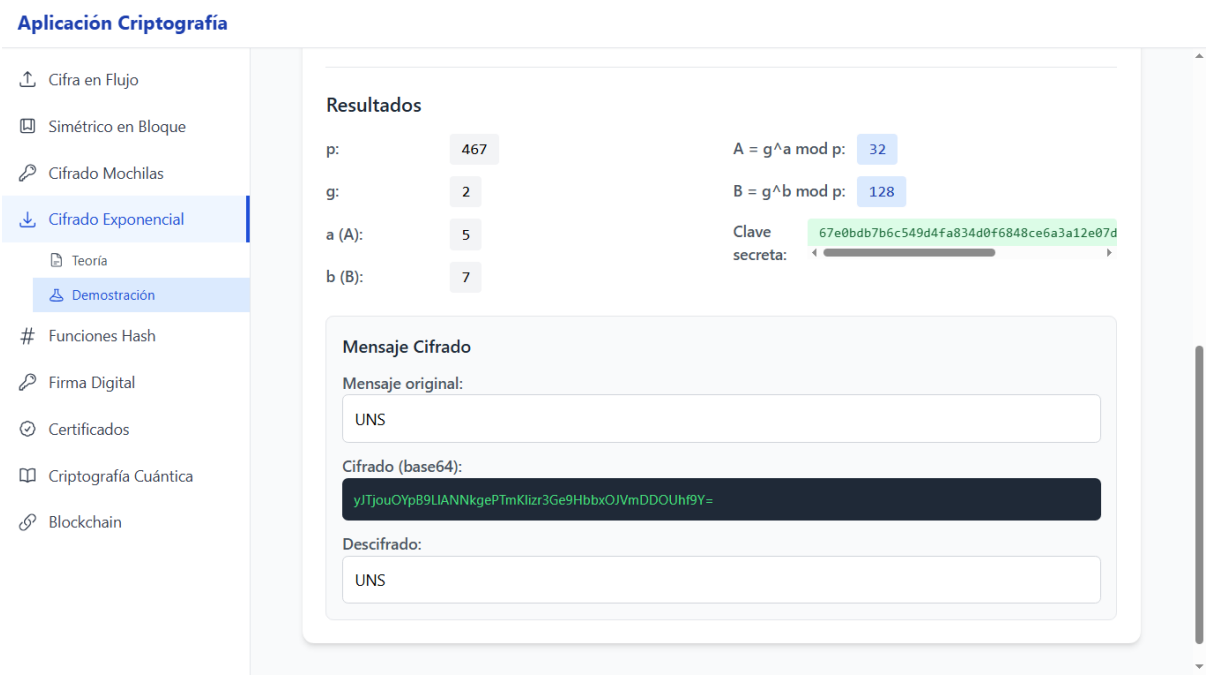


Aplicación Criptografía



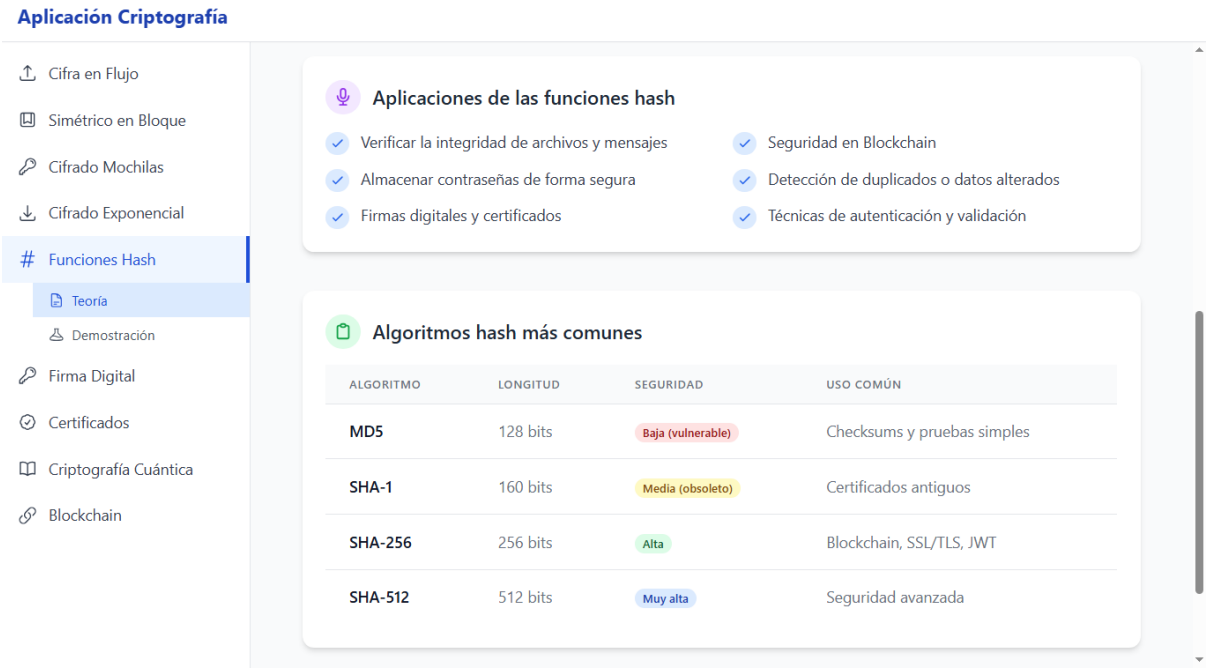
Demostración:





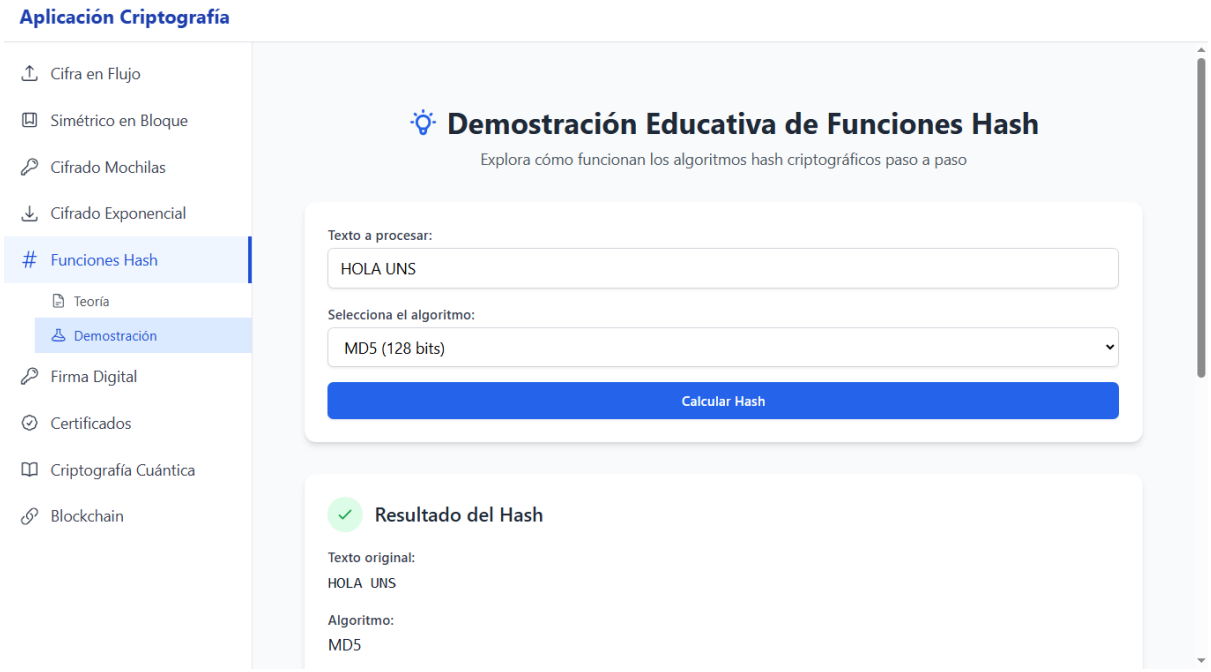
3.6. Funciones Hash

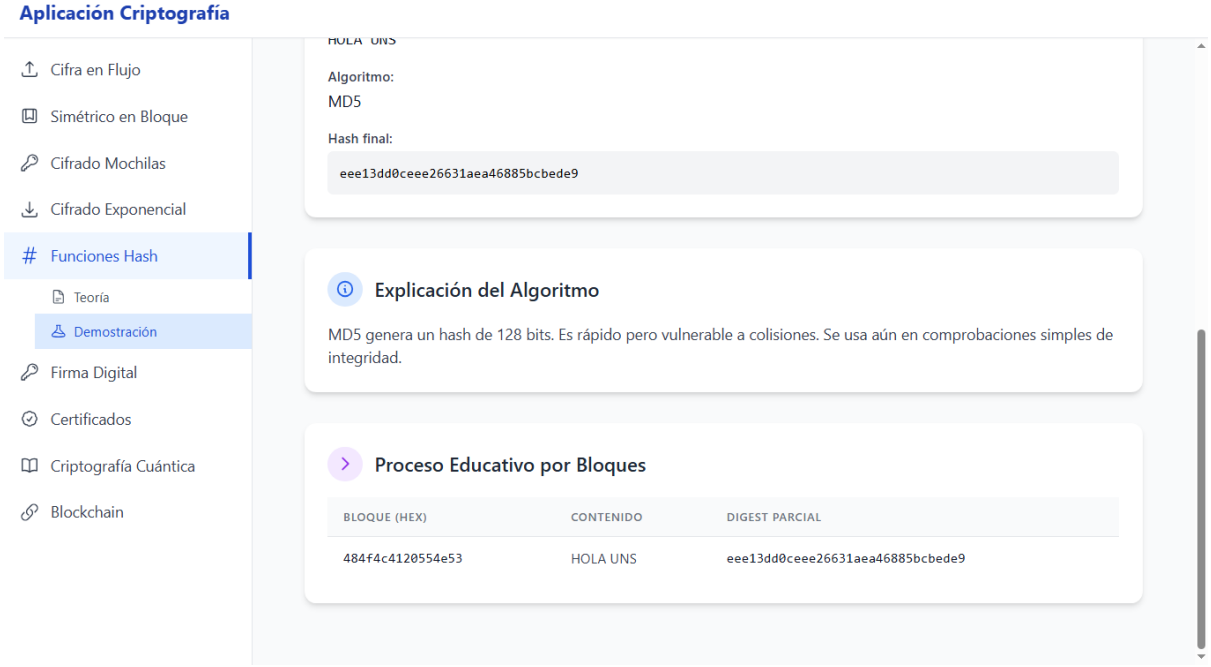
Teoría:



Demostración:

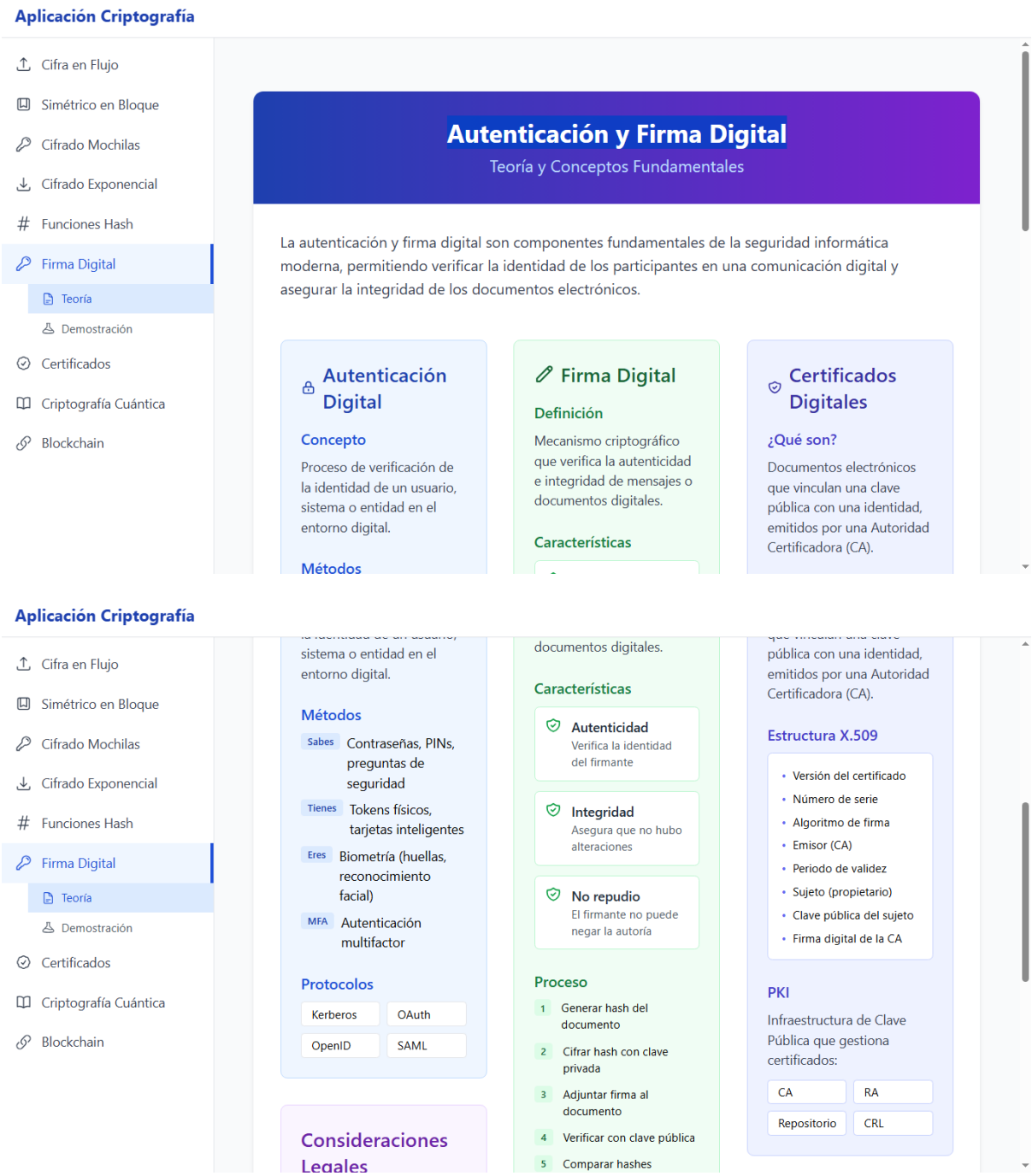
Ingrese un mensaje y seleccione cualquiera de los métodos hash.





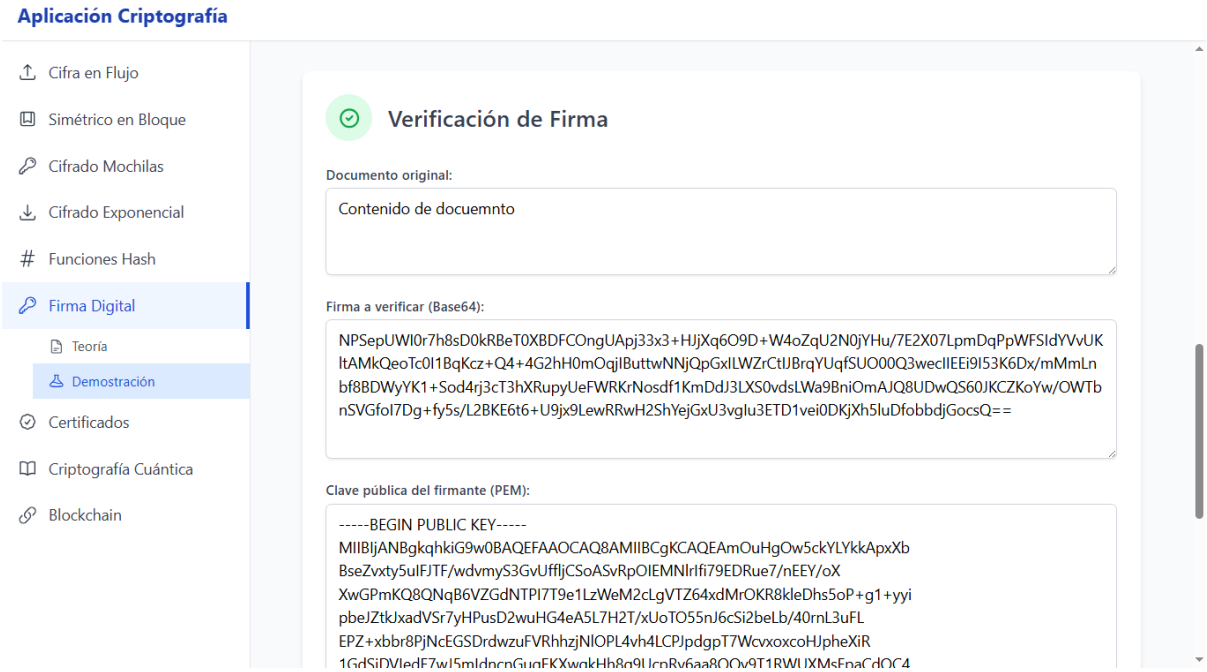
3.7. Autenticación y Firma Digital

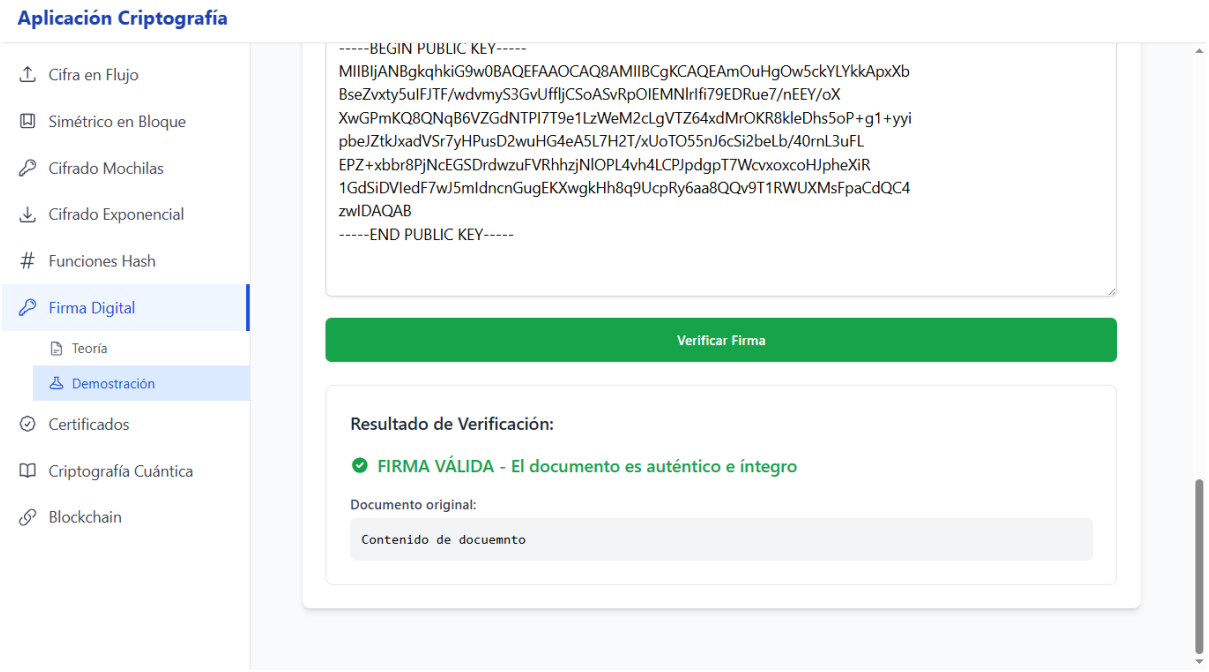
Teoría:



Demostración:


Aplicación Criptografía
Cifra en Flujo
Simétrico en Bloque
Cifrado Mochilas
Cifrado Exponencial
Funciones Hash
Firma Digital
Teoría
Demostración
Certificados
Criptografía Cuántica
Blockchain
Firma digital (Base64):
NPSepUWI0r7h8sD0kRBeT0XBDFCOngUApj33x3+HJjXq6O9D+W4oZqU2N0jYHu/7E2X07LpmDqPpWFSIdYVvUK1tAMkQeoTc0I
Clave pública PEM:
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAmOuHgOw5ckYLYkkApxXb
BseZvxty5uIFJTF/wdvmyS3GvUffljCSoASvRpOIEMNlrIfi79EDRue7/nEEY/oX
XwGPmKQ8QNqB6VZGdNTPI7T9e1LzWeM2cLgVTZ64xdMrOKR8kleDhs5oP+g1+yyi
pbeJZtkJxadVSr7yHPusD2wuHG4eA5L7H2T/xUoTO55nJ6cSi2beLb/40rnL3uFL
EPZ+xbbr8PjNcEGSDrdwzuFVRhhzjNlOPL4vh4LCPJpdgpT7WcvxoxcoHJpheXiR
1GdSiDVIedF7wJ5mIdncnGugEKXwgkHh8q9UcpRy6aa8QQv9T1RWUXMsFpaCdQC4
zwIDAQAB
-----END PUBLIC KEY-----
Verificación
Verificación de Firma
Documento original:
Contenido de docuemnto

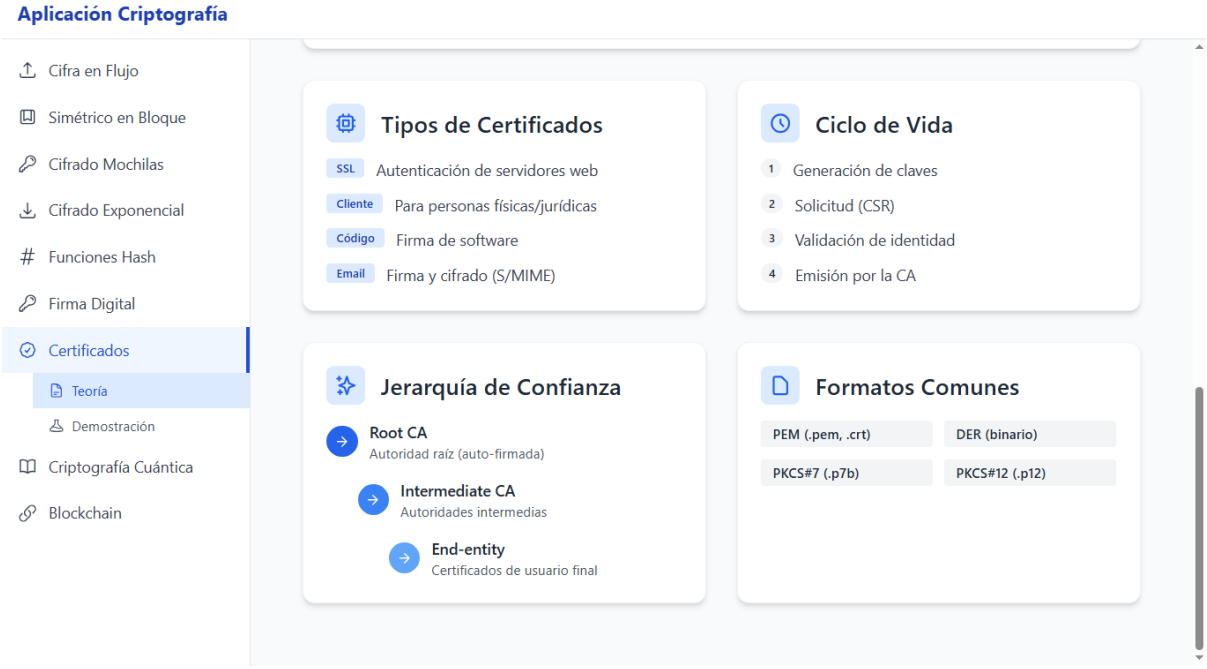

Aplicación Criptografía
Cifra en Flujo
Simétrico en Bloque
Cifrado Mochilas
Cifrado Exponencial
Funciones Hash
Firma Digital
Teoría
Demostración
Certificados
Criptografía Cuántica
Blockchain
Verificación de Firma
Documento original:
Contenido de docuemnto
Firma a verificar (Base64):
NPSepUWI0r7h8sD0kRBeT0XBDFCOngUApj33x3+HJjXq6O9D+W4oZqU2N0jYHu/7E2X07LpmDqPpWFSIdYVvUK
ltAMkQeoTc0I1BqKcz+Q4+4G2hH0mOqjIButtwNNjQpGxILWZrCtIJBrqYUqfSUO00Q3weclIEEi9I53K6Dx/mMmLn
bf8BDWyYK1+Sod4rj3cT3hXRupyUeFWRKrNosdf1KmDdJ3LXS0vdsLWa9BniOmAJQ8UDwQS60JKCZKoYw/OWTb
nSVGfoI7Dg+fy5s/L2BKE6t6+U9jx9LewRRwH2ShYejGxU3vglu3ETD1vei0DKjXh5luDfobbdjGocsQ==
Clave pública del firmante (PEM):
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAmOuHgOw5ckYLYkkApxXb
BseZvxty5uIFJTF/wdvmyS3GvUffljCSoASvRpOIEMNlrIfi79EDRue7/nEEY/oX
XwGPmKQ8QNqB6VZGdNTPI7T9e1LzWeM2cLgVTZ64xdMrOKR8kleDhs5oP+g1+yyi
pbeJZtkJxadVSr7yHPusD2wuHG4eA5L7H2T/xUoTO55nJ6cSi2beLb/40rnL3uFL
EPZ+xbbr8PjNcEGSDrdwzuFVRhhzjNlOPL4vh4LCPJpdgpT7WcvxoxcoHJpheXiR



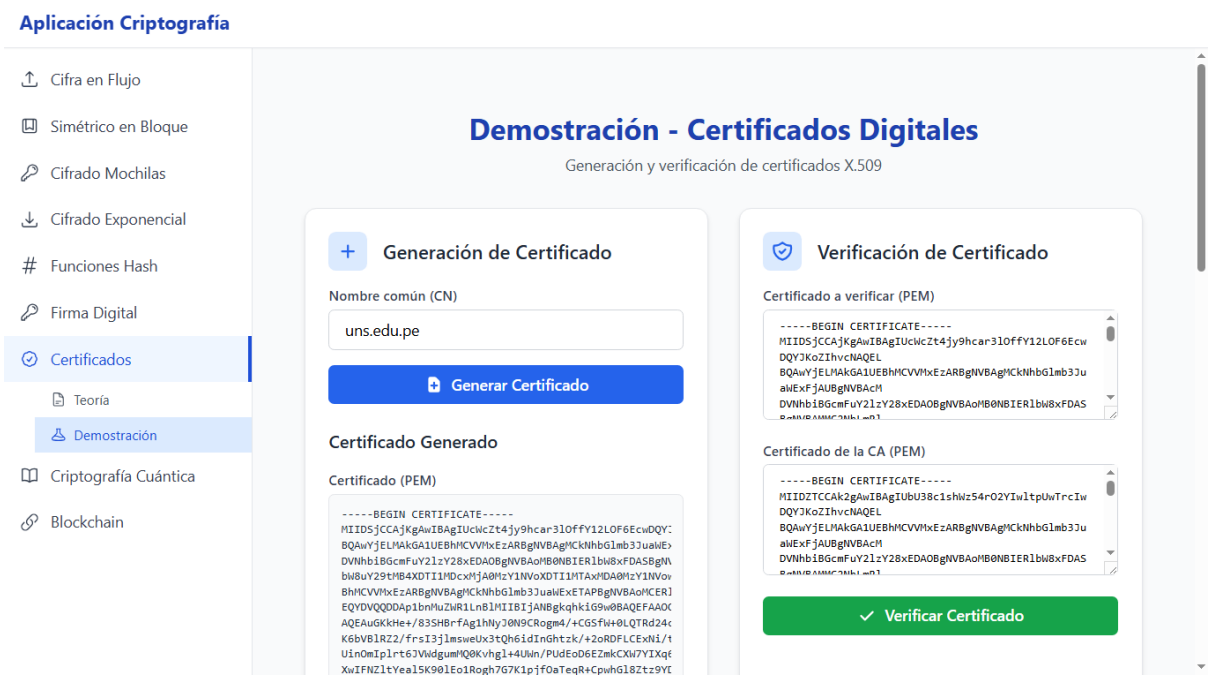
3.8. Certificados Digitales

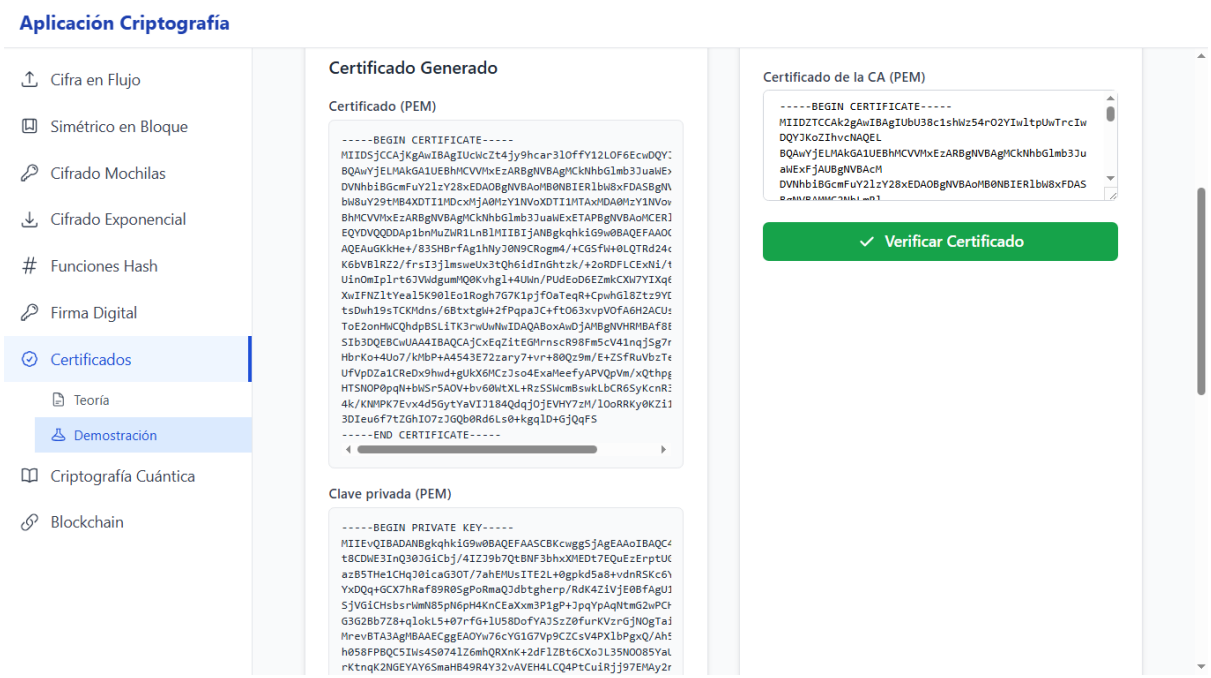
Teoría:


Aplicación Criptografía
Cifra en Flujo
Simétrico en Bloque
Cifrado Mochilas
Cifrado Exponencial
Funciones Hash
Firma Digital
Certificados
Teoría
Demostración
Criptografía Cuántica
Blockchain
Infraestructura de Clave Pública (PKI)
Componentes Principales
CA: Autoridad Certificadora
RA: Autoridad de Registro
Repositorio: Certificados válidos
Mecanismos de Revocación
CRL: Lista de Certificados Revocados
OCSP: Verificación en tiempo real
Tipos de Certificados
SSL Autenticación de servidores web
Cliente Para personas físicas/jurídicas
Código Firma de software
Email Firma y cifrado (S/MIME)
Ciclo de Vida
1 Generación de claves
2 Solicitud (CSR)
3 Validación de identidad
4 Emisión por la CA
Jerarquía de Confianza
Formatos Comunes

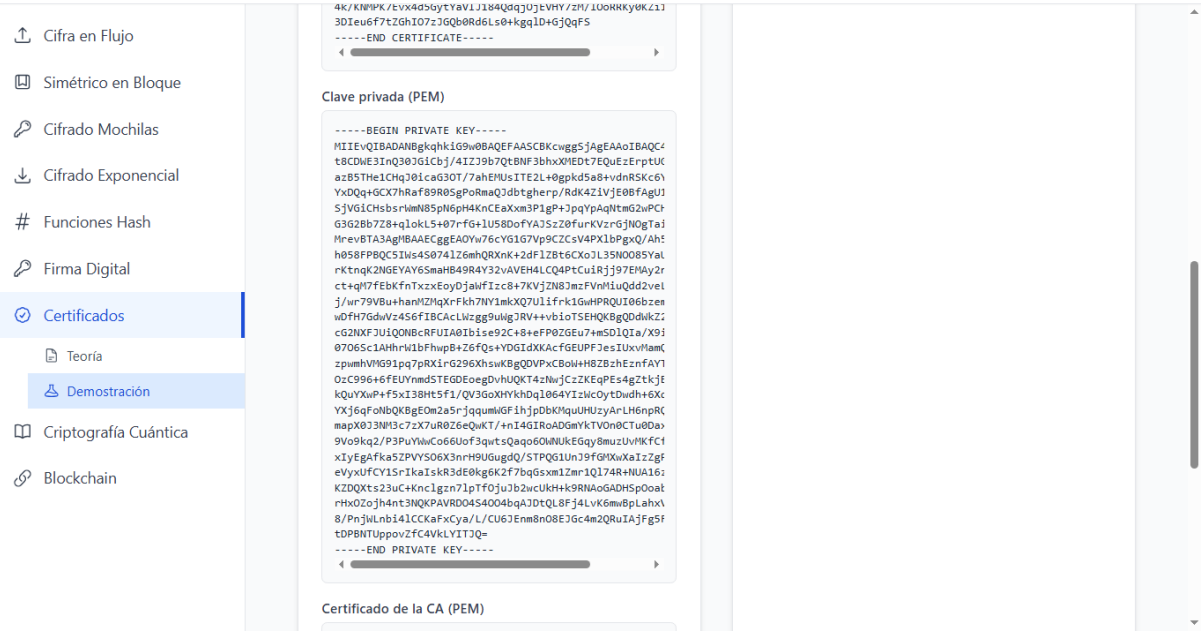

Aplicación Criptografía
Cifra en Flujo
Simétrico en Bloque
Cifrado Mochilas
Cifrado Exponencial
Funciones Hash
Firma Digital
Certificados
Teoría
Demostración
Criptografía Cuántica
Blockchain
Tipos de Certificados
SSL Autenticación de servidores web
Cliente Para personas físicas/jurídicas
Código Firma de software
Email Firma y cifrado (S/MIME)
Ciclo de Vida
1 Generación de claves
2 Solicitud (CSR)
3 Validación de identidad
4 Emisión por la CA
Jerarquía de Confianza
Root CA
Autoridad raíz (auto-firmada)
Intermediate CA
Autoridades intermedias
End-entity
Certificados de usuario final
Formatos Comunes
PEM (.pem, .crt)
DER (binario)
PKCS#7 (.p7b)
PKCS#12 (.p12)

Demostración:

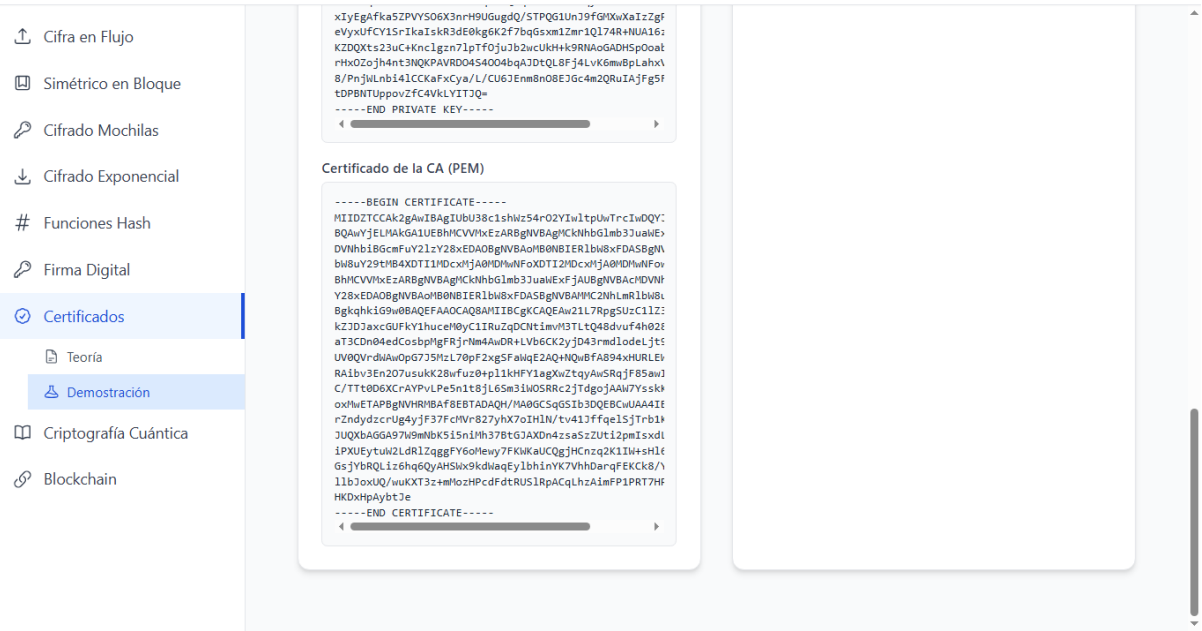




Aplicación Criptografía



Aplicación Criptografía



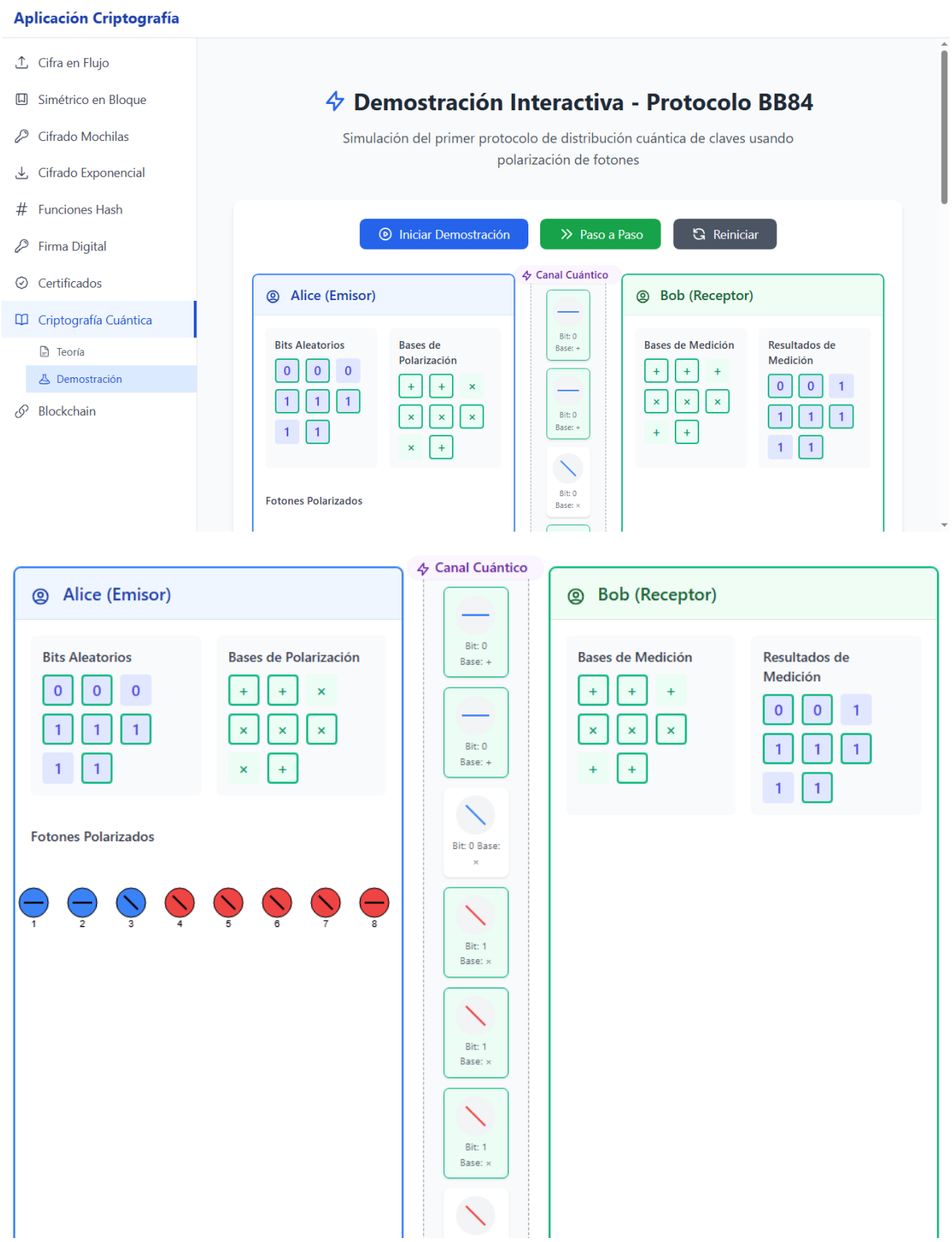
3.9. Criptografía Cuántica

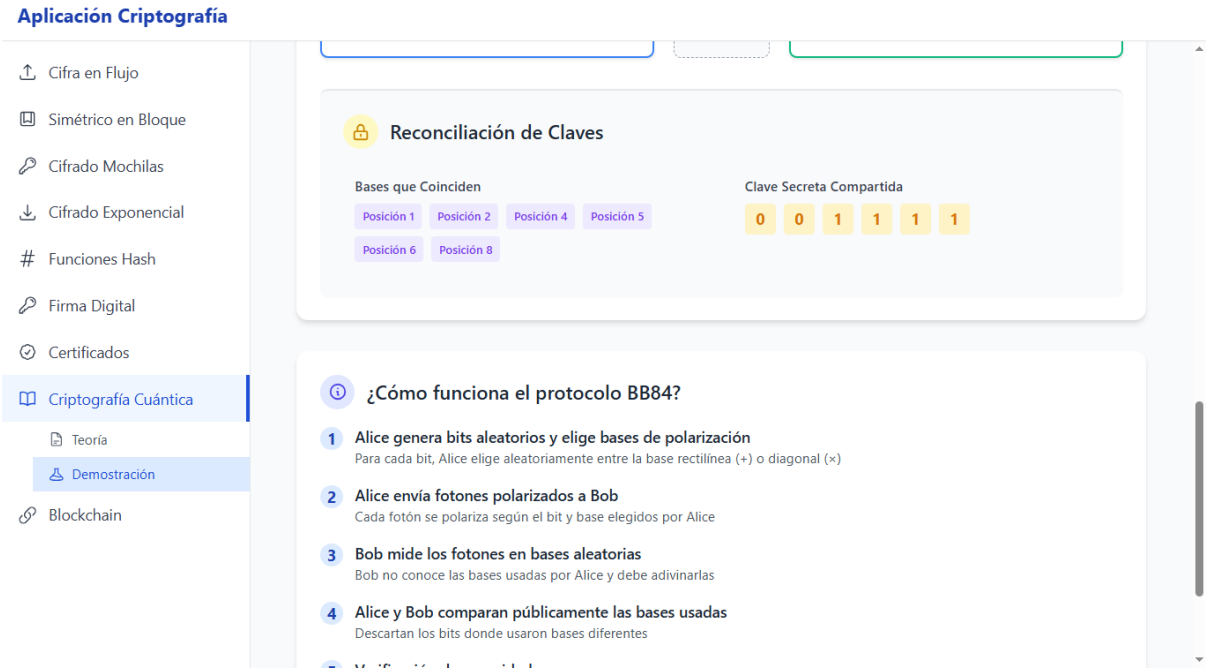
Teoría:

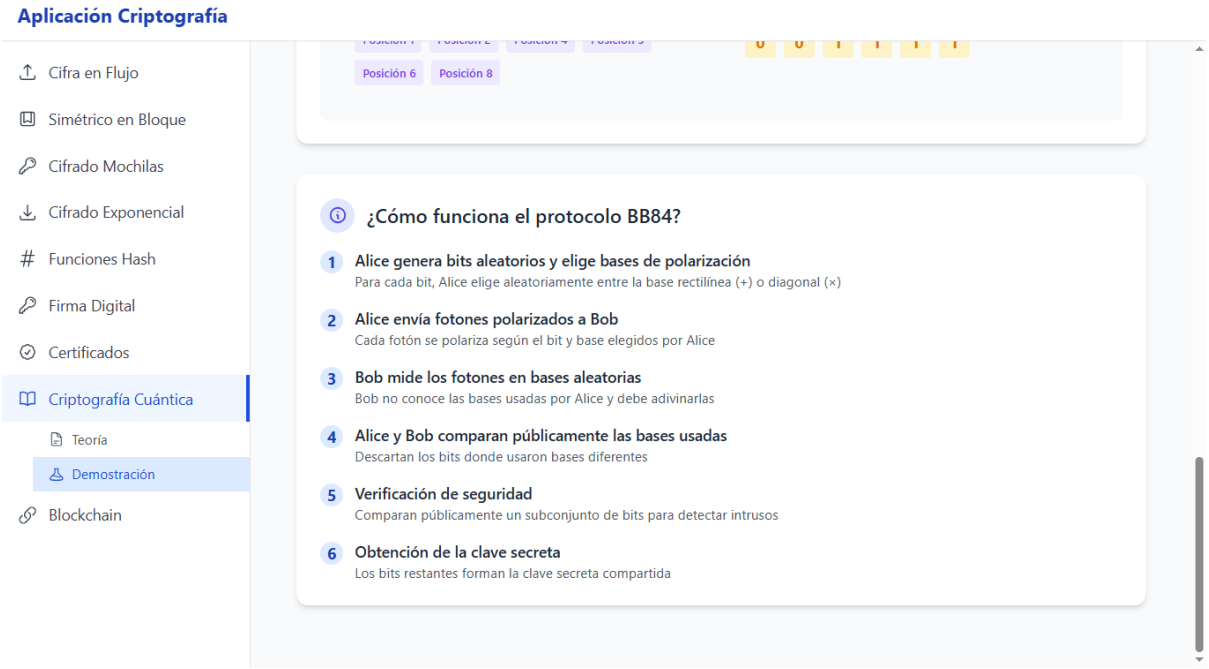

Aplicación Criptografía
Cifra en Flujo
Simétrico en Bloque
Cifrado Mochilas
Cifrado Exponencial
Funciones Hash
Firma Digital
Certificados
Criptografía Cuántica
Teoría
Demostración
Blockchain
BB84 (1984)
Desarrollado por Charles Bennett y Gilles Brassard, fue el primer protocolo práctico de distribución cuántica de claves.
Utiliza estados de polarización de fotones en dos bases diferentes (rectilínea y diagonal)
La seguridad deriva del principio de incertidumbre: medir en la base incorrecta altera el estado
E91 (1991)
Protocolo diseñado por Artur Ekert que utiliza pares de fotones entrelazados para establecer claves secretas.
Aprovecha las correlaciones cuánticas entre partículas entrelazadas
La violación de las desigualdades de Bell detecta cualquier intento de interceptación
Teorías Relacionadas
Teoría de la Información Cuántica
Extiende los conceptos clásicos de bits a qubits, introduciendo superposición y entrelazamiento como recursos computacionales.
Computación Cuántica
Utiliza fenómenos cuánticos para realizar operaciones que superan exponencialmente las capacidades de las computadoras tradicionales.
Teleportación Cuántica
Protocolo que transfiere estados cuánticos entre ubicaciones distantes usando entrelazamiento y comunicación clásica.


Aplicación Criptografía
Cifra en Flujo
Simétrico en Bloque
Cifrado Mochilas
Cifrado Exponencial
Funciones Hash
Firma Digital
Certificados
Criptografía Cuántica
Teoría
Demostración
Blockchain
Teorías Relacionadas
Teoría de la Información Cuántica
Extiende los conceptos clásicos de bits a qubits, introduciendo superposición y entrelazamiento como recursos computacionales.
Computación Cuántica
Utiliza fenómenos cuánticos para realizar operaciones que superan exponencialmente las capacidades de las computadoras tradicionales.
Teleportación Cuántica
Protocolo que transfiere estados cuánticos entre ubicaciones distantes usando entrelazamiento y comunicación clásica.
Aplicaciones Prácticas
Distribución Cuántica de Claves (QKD)
Sistemas comerciales ya disponibles para transmisión segura de claves criptográficas
Redes de Comunicación Seguras
Implementaciones en redes bancarias, gubernamentales y militares
Protección Post-Cuántica
Defensa contra futuros ataques con computadoras cuánticas
Desafíos Actuales
Limitaciones de Distancia
La atenuación en fibras ópticas limita la transmisión a ~100-200 km sin repetidores
Repeaters Cuánticos
La tecnología para repetidores cuánticos fiables aún está en desarrollo
Implementación Práctica
Vulnerabilidades en dispositivos reales pueden explotarse en ataques tipo "side-channel"

Demostración:



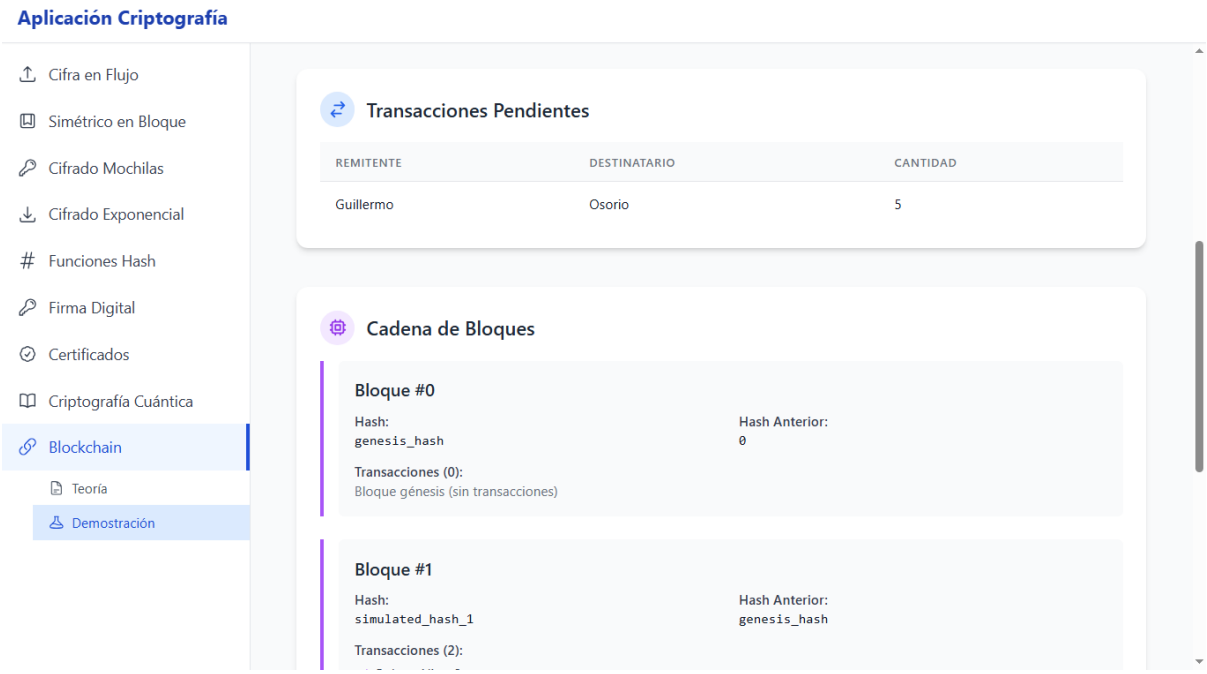

Aplicación Criptografía
Cifra en Flujo
Simétrico en Bloque
Cifrado Mochilas
Cifrado Exponencial
Funciones Hash
Firma Digital
Certificados
Criptografía Cuántica
Teoría
Demostración
Blockchain
Reconciliación de Claves
Bases que Coinciden
Posición 1
Posición 2
Posición 4
Posición 5
Posición 6
Posición 8
Clave Secreta Compartida
0 0 1 1 1 1
¿Cómo funciona el protocolo BB84?
1 Alice genera bits aleatorios y elige bases de polarización
Para cada bit, Alice elige aleatoriamente entre la base rectilínea (+) o diagonal (×)
2 Alice envía fotones polarizados a Bob
Cada fotón se polariza según el bit y base elegidos por Alice
3 Bob mide los fotones en bases aleatorias
Bob no conoce las bases usadas por Alice y debe adivinarlas
4 Alice y Bob comparan públicamente las bases usadas
Descartan los bits donde usaron bases diferentes


Aplicación Criptografía
Cifra en Flujo
Simétrico en Bloque
Cifrado Mochilas
Cifrado Exponencial
Funciones Hash
Firma Digital
Certificados
Criptografía Cuántica
Teoría
Demostración
Blockchain
Posición 6
Posición 8
¿Cómo funciona el protocolo BB84?
1 Alice genera bits aleatorios y elige bases de polarización
Para cada bit, Alice elige aleatoriamente entre la base rectilínea (+) o diagonal (×)
2 Alice envía fotones polarizados a Bob
Cada fotón se polariza según el bit y base elegidos por Alice
3 Bob mide los fotones en bases aleatorias
Bob no conoce las bases usadas por Alice y debe adivinarlas
4 Alice y Bob comparan públicamente las bases usadas
Descartan los bits donde usaron bases diferentes
5 Verificación de seguridad
Comparan públicamente un subconjunto de bits para detectar intrusos
6 Obtención de la clave secreta
Los bits restantes forman la clave secreta compartida

3.10. Blockchain

Teoría:

Demostración:


Aplicación Criptografía
Cifra en Flujo
Simétrico en Bloque
Cifrado Mochilas
Cifrado Exponencial
Funciones Hash
Firma Digital
Certificados
Criptografía Cuántica
Blockchain
Teoría
Demostración
Transacciones Pendientes
REMITENTE
DESTINATARIO
CANTIDAD
Guillermo
Osorio
5
Cadena de Bloques
Bloque #0
Hash:
genesis_hash
Hash Anterior:
0
Transacciones (0):
Bloque génesis (sin transacciones)
Bloque #1
Hash:
simulated_hash_1
Hash Anterior:
genesis_hash
Transacciones (2):


Aplicación Criptografía
Cifra en Flujo
Simétrico en Bloque
Cifrado Mochilas
Cifrado Exponencial
Funciones Hash
Firma Digital
Certificados
Criptografía Cuántica
Blockchain
Teoría
Demostración
Bloque #1
Hash:
simulated_hash_1
Hash Anterior:
genesis_hash
Transacciones (2):
Bob → Alice: 2
Guillermo → Osorio: 5
¿Cómo funciona esta demostración?
1 Transacciones pendientes
Las transacciones se agregan al pool de transacciones pendientes antes de ser incluidas en un bloque.
2 Minería de bloques
Al minar, las transacciones pendientes se incluyen en un nuevo bloque que se añade a la cadena.
3 Encadenamiento criptográfico
Cada bloque referencia al anterior mediante su hash, creando una cadena inmutable.
4 Inmutabilidad
Una vez añadido, un bloque no puede alterarse sin invalidar todos los bloques posteriores.

IV. CONCLUSIONES

El desarrollo del software que recopila diferentes métodos de criptografía permitió consolidar un entendimiento profundo de los principios teóricos y prácticos de las técnicas de seguridad informática abordadas en el proyecto. A través de la implementación de módulos como cifrado en flujo, cifrado simétrico en bloque, cifrado asimétrico (mochilas y exponencial), funciones hash, autenticación y firma digital, certificados digitales, criptografía cuántica y blockchain, se logró demostrar la importancia y aplicabilidad de estas herramientas en la protección de la información.

Entre los hallazgos más relevantes se destaca:

- Eficiencia y versatilidad: Los algoritmos de cifrado simétrico (como AES) resultaron óptimos para proteger datos en tiempo real, mientras que los métodos asimétricos (como RSA) demostraron su utilidad en el intercambio seguro de claves.
- Integridad y autenticidad: Las funciones hash y las firmas digitales son fundamentales para garantizar la autenticidad de los mensajes y la detección de alteraciones.
- Futuro de la criptografía: Tecnologías emergentes como la criptografía cuántica y blockchain presentan un potencial disruptivo para resolver problemas de seguridad tradicionales, aunque su adopción aún enfrenta desafíos técnicos.